Device Utility Programs Manual

August 1984

Table of Contents

27 August 1984

1. Introduction

This manual describes the Device Programs that run under Accent with the S5 version of the system. There are 3 types of PERQs at CMU:

PERQ1A   have 24 mb Shugart 14" disks (any references to PERQ1s are really to PERQ1as.)

PERQ2   have 35 mb Micropolis 8" disks.

PERQt2   have 5.25" disks of various capacities.

All PERQs on the 3MHz Ethernet are PERQ1as running the OIO microcode. All PERQs on the 10MHz Ethernet are either PERQ1as running CIO, PERQ2s or PERQt2s. See the <u>Introduction to the Spice User's Manual</u> or send mail to Spice@Spice to determine what type of PERQ you are using.

2. Bindboot

Bindboot attaches a boot file to a boot character. A boot character is a lower-case letter on the keyboard which brings up a particular operating system; the boot key is pressed after the workstation is turned on and a black pattern appears briefly on the screen.

There are two types of boot files. The system file (standard extension .Boot) contains the operating system code. The Interpreter file (standard extension .MBoot) contains the microcode for the workstation. The system file is constant across various models of the PERQ workstation, but there is a specific Interpreter for each type of workstation.

Format:
```
BINDBOOT {-switches}
```

Switches are:

<u>Switch</u>   <u>Description</u>

-HELP   Displays information about the Bindboot facility

-SYSTEM = \<FILENAME\>
                 Uses \<filename\> as the system boot file

-INTERPRETER = \<FILENAME\>
                 Uses \<filename\> as the interpreter boot file.

-BOOTCHARACTER = \<Char\>
                 Binds the boot to the specified character.

-NOSYSTEM   Writes only the interpreter boot file, not the system boot file.

-NOINTERPRETER
                 Writes only the system boot file, not the interpreter boot file.

If you do not supply switches, Bindboot will prompt for the required information.

To bind only the system boot file, supply the following command line:
```
bindboot -BootChar=<c> -System=<filename> -NoInterpreter
```

To bind only the interpreter boot file, supply the following command line:
```
bindboot -BootChar=<c> -Interpreter=<filename> -NoSystem
```

3. Floppy

The FLOPPY utility formats, tests, reads, and writes RT-11 format floppy disks. You can use FLOPPY to transfer files between the hard disk and the floppy disk. Note that the RT-11 file format restricts filenames to six characters with a three character extension. Thus, if you wish to copy a file from the hard disk to the floppy disk, Floppy limits the file name on the floppy to six characters and the extension to three characters (you can enter the full file name, but only the first six characters will name the file on the floppy). Valid characters are the upper and lower case alphabetics, the digits 0 through 9, the dollar sign, and the period; Floppy uses the RAD50 character set.

Floppy requires that a FloppyServer be running. If Floppy is invoked and FloppyServer cannot be found, Floppy will abort with the message,

** could not fild FloppyServer at [MachineName] FloppyServer

If the Environment variable "FLOPPYSERVERNAME" is set, Floppy will use that port name in finding the Floppy Server. For example, FLOPPYSERVERNAME could be defined as "[Spicel]FloppyServer" to use the Floppy Server running on the remote PERQ, Spicel. To start a Floppy Server, run "FloppyServer".

Format:
```
FLOPPY [command] {-switch}
```

To execute a single FLOPPY command, enter the command on the command line.

To execute multiple FLOPPY functions, type FLOPPY and press return. In this case, the utility prompts with FLOPPY>. You can then enter commands. All of the Floppy commands prompt for missing arguments.

You can also direct FLOPPY to access a command file containing a list of FLOPPY commands. Simply enter an at sign (@) followed by the file name that contains the FLOPPY commands (@file). The command file must be on the default device (the hard disk).

Some FLOPPY commands require confirmation. This confirmation comes from the keyboard even if a user command file is in use. If you use the FAST command (see below), no confirmation is required. The ZERO and FORMAT commands, however, require an explicit -switch to override the confirmation request. The command file must be on the default device (the hard disk).

While FLOPPY is processing a command, the progress bar in the window title line, as well as the progress bar in the icon, moves from left to right. When it reaches the right side, the operation is complete.

FLOPPY only accepts wildcards for the DELETE and DIRECT commands (see the respective command descriptions). In addition, the only wildcard available is the asterisk (*) and it can only be used by itself in either the filename or extension part of a file specification (or both).

Valid FLOPPY commands and their switches are:

Command	Description
ASK	Require confirmation on each file name. Accepts the -HELP switch.
NOASK	Do not require confirmation on each file name. Accepts the -HELP switch. (default)
COMPARE	This command takes a disk file and a floppy file (in that order) and ensures that all bytes are identical.

Format:
```
COMPARE [diskfile][~][floppyfile]
```

You can specify multiple disk files and multiple floppy files for the comparison. If you specify multiple arguments, you must separate like arguments by a comma (,) and delimit input from output arguments with the tilde character (~). For example,
```
Compare disk1.fil,disk2.fil~flop1.fil,flop2.fil
```

The COMPARE command prints a message for every block that contains a difference.

The command accepts the following switches:

-WAIT   pauses if differences are encountered

-NOWAIT   does not pause

-HELP   provides information on the Compare command

COMPRESS   This command moves files so that all the unused blocks are at the end of the floppy; the command coalesces free space on the floppy.

```
COMPRESS {-switch}
```

Because the procedure takes a long time, the Floppy facility asks you to confirm that you want to compress. To start compression without being asked to confirm your intentions, use the switch -NOCONFIRM.

The command does not accept input or output arguments, but has a switch that turns verification on or off. The default is verify--COMPRESS checks every transfer to assure there are no errors. The -NOVERIFY switch overrides the default.

The command also accepts the -HELP switch.

Do not interrupt this command once it starts; interrupting the Compress command renders the floppy data unreadable.

CONFIRM   Require confirmation before deleting or overwriting a file (default). Accepts the -HELP switch.

NOCONFIRM   Do not require confirmation before deleting or overwriting a file (much like a global switch). Accepts the -HELP switch.

DELETE   This command deletes a file or multiple files from the floppy.

Format:
```
DELETE [file][...,filen] {-switch}
```

To delete multiple files, you must separate the filenames with a comma (,). By default, when wildcards are used the DELETE command requests confirmation before deleting a file. You can override this by specifying the -NOCONFIRM switch.

You can specify an asterisk (*) only in place of the file name and/or in place of the file extension. For example, *.txt, *.*, or myprogram.* are valid uses of the wildcard (you cannot specify my*.*).

The command also accepts the -HELP switch.

DENSITY   This command displays whether the floppy is single or double density.

Format:
```
DENSITY
```

27 August 1984

The command accepts no arguments nor switches except -HELP.

DIRECTORY   This command lists the files contained on a floppy and optionally writes the directory listing to a file on the hard disk.

Format:
```
DIRECTORY [file][~][outputfile] {-switch}
```

If you specify the DIRECTORY command with no arguments, it lists all the files contained on the floppy. If you specify an input argument, DIRECTORY lists all files that match the specified filename.

By default, the DIRECTORY command prints a full listing. To override this and print only the file names, specify the -SHORT switch.

If you specify an output argument, DIRECTORY only writes the file names (-SHORT is the default when you specify an output file) to a hard disk file with that name.

Note that when you direct the output to a file, Directory does not display the filenames on the screen.

The command accepts the -HELP switch.

FAST   This command turns off requests for confirmation for all subsequent commands except Format and Zero; the command sets the -NOASK and -NOCONFIRM switches for all subsequent commands except Format and Zero.

Format:
```
FAST
```
The command accepts no arguments or switches except -HELP.

FORMAT   This command formats a floppy disk and destroys its current contents. Format command switches permit you to specify the floppy density, number of sides, and whether or not to test after formatting. The -Interleave switch permits you to specify the number of floppy sectors between sequentially numbered sectors. For example, if the -Interleave value is one, the sectors are numbered sequentially. If the -Interleave value is two, every other sector is numbered sequentially.

Format:
```
FORMAT {-switch}
```

The switches are:

-DblDensity -SingleDensity (default) -Noconfirm -Singlesided -DblSided (default) -Test -NoTest (default) -Interleave = value (default is 2) -Help

GET   This command copies one or more floppy files to the hard disk.

Format:

```
GET [floppyfile][~][hardfile] {-switch}
```

In its simplest form, you specify only an input filename to copy from the floppy to the hard disk. GET then copies the floppy file to the hard disk using the same filename. If the filename already exists on the hard disk, GET requests confirmation before overwriting it. If you omit the input file name, GET prompts for the file to copy. Other forms of the command permit you to name the hard disk file; for example:

```
GET floppyfile~harddiskfile
```

To specify multiple files, use the following construct:

```
GET f1,f2,...fn~h1,h2,...hn
```

Like the COMPRESS command, GET takes the -ASK, -NOASK, -VERIFY and -NOVERIFY switches, as well as -HELP.

The GET command requires confirmation before overwriting a file on the hard disk. You can specify the -NOCONFIRM switch to override this action. If the harddisk filename, as given, is longer than RT-11 format (as previously described), GET will automatically truncate the floppy filename to the standard size. For example, in the case where the file "accent.pas" is on the floppy, and the command "get accenttype.pasmac" is given, accent.pas will be copied to the harddisk as accenttype.pasmac. This, of course, could lead to accidental overwriting. It is preferable therefore to use the form "get accent.pas ~ accenttype.pasmac"

HELP

When issued as a command, HELP provides general information for the FLOPPY utility. You can get specific help by using the -HELP switch. Note that HELP and -HELP override any other commands or switches; FLOPPY displays the requested help and then reprompts.

PATH

This command changes the current hard disk directory.

Format:

```
Path [pathname]
```

Specify the directory for the new path. If you omit the directory, Path prompts for it. The command accepts the -HELP switch.

PAUSE

This command suspends FLOPPY execution.

Format:

```
PAUSE
```

Press carriage return to continue.

The command accepts the -HELP switch.

27 August 1984

PUT   This command parallels the GET command, but transfers a file or files from the hard disk to the floppy disk.

Format:
```
PUT [hardfile][~][floppyfile] {-switch}
```

PUT also requires confirmation before overwriting an existing file on the floppy (override this with -NOCONFIRM) and, like GET, prompts for a missing file name and accepts the -ASK, -NOASK, -VERIFY and -NOVERIFY switches, as well as -HELP.

In a manner similar to that of GET, PUT will truncate the floppy filename if the harddisk filename is too long. For example, the command "put accenttype.pasmac" will copy the harddisk file onto the floppy as "accent.pas". Again, this could lead to accidental overwriting. The preferred form of the command is "put accenttype.pasmac ~ acctyp.pas".

QUIT·   This command exits the FLOPPY utility.

Format:
```
QUIT
```

It accepts the -HELP switch.

RENAME   This command changes the name of a floppy file.

Format:
```
RENAME [oldname][~][newname][-NOCONFIRM]
```

If the new filename already exists, RENAME asks for confirmation before overwriting it. You can override this with the -NOCONFIRM switch. The command also accepts the -HELP switch.

SAFE   ASK only on wildcards (default). Accepts the -HELP switch.

TYPE   This command displays a floppy file on the screen. A progress bar moves from left to right in the window title line, as well as in the icon, to show how much of the file has been displayed.

Format:
```
TYPE [filename]
```

It accepts the -HELP switch.

VERIFY   Verify correctness of data transfers. Accepts the -HELP switch.

NOVERIFY Do not verify correctness of transfers. Accepts the -HELP switch.

ZERO   This command creates a new directory on the floppy. ·

Format:

```
ZERO {-switch}
```

By default, the directory matches the number of sides on the floppy. You can override this by specifying the -SINGLESIDED switch. The Zero command always requests confirmation before creating the directory. You can override this only by specifying the -NOCONFIRM switch. In the process of creating the new directory, the ZERO command destroys the contents of the current floppy. Use the ZERO command after formatting a floppy (see the FLOPPY FORMAT command description) or whenever you wish to destroy the current content of a floppy.

Note that when you issue the ZERO command, you irrevocably destroy the current contents.

It is usually necessary to run FORMAT when using a new floppy. Whenever you use FORMAT, you must also ZERO. When re-using a floppy disk, it may not be necessary to format if you simply want to delete all existing files; use ZERO.

The command accepts the -HELP switch.

4. Partition

Partition creates new partitions or modifies the names and sizes of existing partitions on a disk. Most uses of Partition include initializing pages, which destroys all old data in the partition's physically contigious disk area.

The file system restricts partitions on a disk to fewer than 64K logical blocks [pages]. However, all partitions start and end on cylinder boundaries, and the Partition program enforces this constraint. The maximum size allowed is 65,040 blocks. The maximum number of partitions allowed is 15. Note however, that the maximum usable size for POS is 32520 blocks and the maximum number of partitions allowed by POS is 10 (in the first 31MB of disk addressed by POS).

Format:
```
PARTITION [-HELP] [-DEBUG]
```

The -Help switch gives information about the Partition program. The -Debug switch is a learning tool that leads you through the series of questions that the Partition program asks, without making any changes to your disk.

Partition may be used on any one of the multiple disks (units 0-3) that may be mounted on your PERQ. Partition can create new partitions on unused portions of a disk. For existing partitions, Partition can split, merge, rename or initialize. In addition, Partition can rename the whole device.

27 August 1984

There are several points to keep in mind when using Partition:

- You can set free list address schemes to be device-relative or partition-relative. Device-relative free lists are satisfactory for most partitons. However, if a partition includes a logical block number >64K, the free list must be partition-relative because a link in the free list chain cannot hold an address >64K.

- If you change a free list from device-relative to partition-relative (or vice versa), you must initialize the pages in that partition. Initializing the pages rewrites the links in the free list chain in the current relative manner.

- Initializing pages also rewrites each data block within a partition, and destroys any data that was in the block.

- Initializing a partition rewrites header information in the Partition Information Block. If you initialize the partition, you also should initialize the pages within it.

- You should not run Partition on the partition you are currently pathed to, or on partitions accessed by the operating system (such as your boot, spice, and paging partitions).

- You should not abort Partition by using CNTL-Del C or CNTL-Del K once Partition has begun writing to the disk.

All of the questions that Partition asks are listed on the following pages, with explanatory text for each one. Note that your responses control the logical flow of the questions. Depending on your responses, not all the questions appear. The flow of questions is shown in Figure 1.

1. Do you want to debug? (does not do any writes) [No]

    The response to this question specifies whether or not the Partition program actually initializes or modifies a disk.

    Replying Yes is the same as specifying the -Debug switch. (If you specified the switch, you will not receive this question.) If you respond Yes, the program simply continues with the question sequence, but the disk is not modified. Respond Yes to familiarize yourself with the logical flow of Partition questions and to prepare your responses.

    If you respond No, the program modifies the disk based on your responses to succeeding queries.

    The default response is No (the program modifies the disk).

    In case you do not remember whether you gave the -Debug switch or how you replied to this question, the window title line shows "Debug ON" if you are only running through the questions or "Debug OFF" if you are actually making changes to the disk.

2. Which unit do you wish to Partition (0-3)? [0]

    Respond with the unit number of the disk you wish to partition. The default is Unit 0. Use the default if there is only one disk in your workstation.

    Depending on the type of disk in your workstation, the next question will be 3a, 3b, or 3c.

3. You will be asked one of the following questions:

    a. This seems to be a MICROPOLIS 8-inch disk. Is this right? [Yes]

    b. This seems to be a SHUGART 14-inch disk. Is this right? [Yes]

    c. This seems to be a 5.25" disk. Is this right? [Yes]

    The above questions confirm the type of disk. The next question is 4.

4. Do you want to rename the device? [No]

    This question begins the disk modification sequence. This first question in the sequence allows you to change the name of the disk.

    Before renaming a disk, note that the POS linker incorporates the name of the disk (device) where a program was linked into the Runfile. Therefore, if you rename the disk, you cannot run existing POS programs on that disk. This includes the Shell, Login, and the Partition program itself.

    If you respond Yes, Partition requests confirmation and then prompts for the new disk name. See question 5.

    If you respond No, the program asks whether or not you want to partition the remainder of the disk. See question 6.

    The default response to this question is No (to not rename the disk).

5. New device name [current name]

    If you responded Yes to question 4 and to the subsequent request for confirmation, this question allows you to enter a new disk name. Enter a disk name from 1 to 8 characters. The Partition program writes the name you enter in the Device Information Block (DIB words 114 through 117).

    Note that the default response to this question is the current name of the disk.

27 August 1984

Figure 1: Flow of Partition Questions
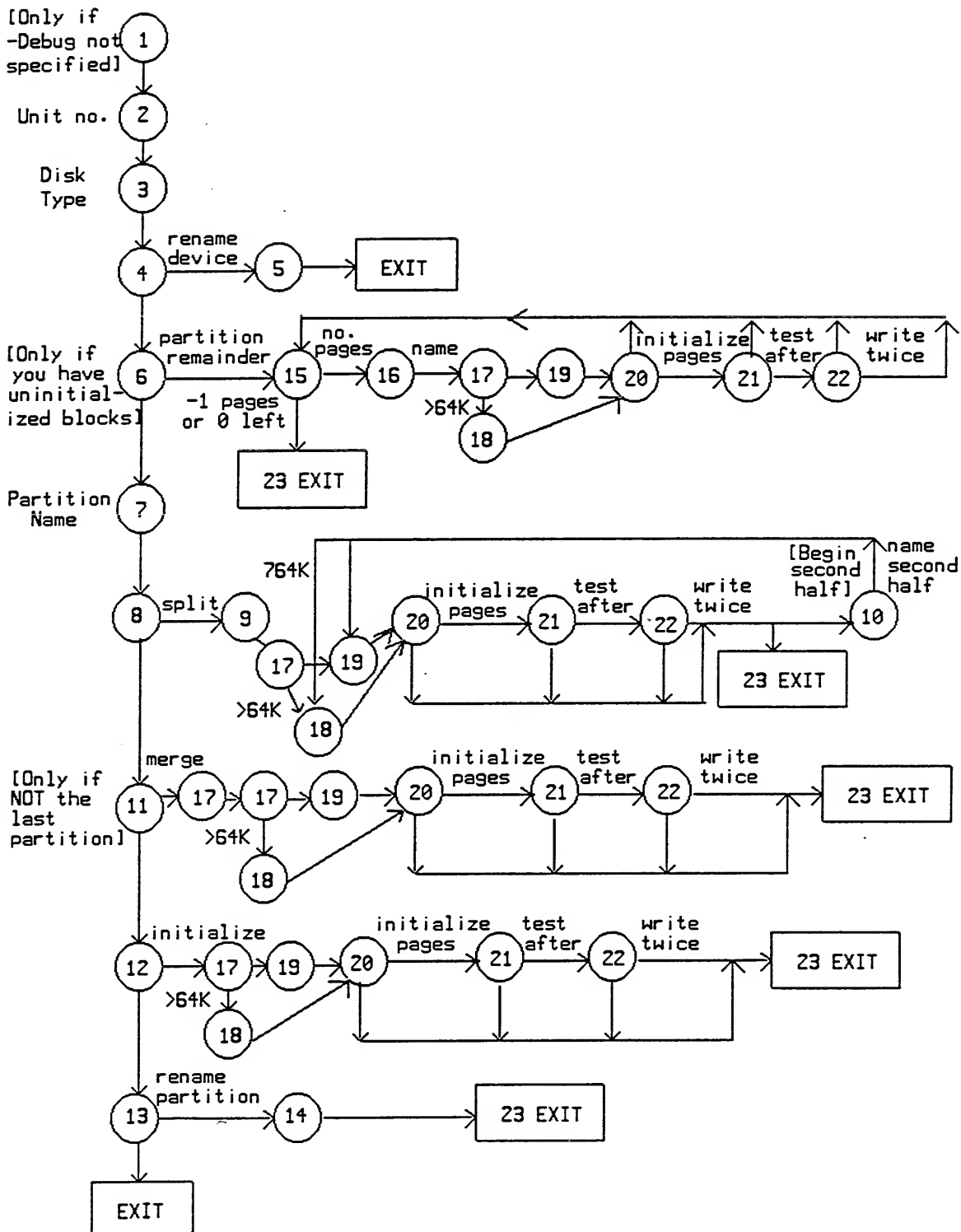
6. Do you want to partition the remainder of the device? [No]

This question may be asked if you responded No to question 4. It is *not* asked if you have no uninitialized cylinders on your disk. Respond YES if your disk was not fully initialized and has unused cylinders (at the end).

If you respond Yes to this question, Partition asks for information about each new partition in turn. See questions 15-22. This allows you to create more partitions in the uninitialized area at the end of your disk. You may stop before the end of the disk by answering -1 to question 15.

7. Which partition do you want to modify?

This question is asked if you responded No to question 6. Enter the name of the partition you wish to modify. The program displays a summary of that partition's Partition Information Block and then prompts for the modification. See questions 8 through 13.

There is no default response to this question.

8. Do you want to split this partition? [No]

Since files cannot cross partition boundaries, it is never a good idea to split a partition that contains files. The system destroys any file that has blocks in both partitions. However, it is safe to split an empty partition or to split a partition that you plan to erase.

If you respond Yes to this question, Partition requests confirmation and then prompts for the number of pages in each partition and the name of the new partition. See questions 9 and 10.

If you respond No to this question, Partition continues with the modification queries. See question 11.

9. How many pages would you like in the first half?

The response to this question specifies how much of the partition to leave with the old partition (the partition being split). The remainder of its pages are put in the new partition. The Partition program then asks if you want to initialize the partition pages and thus create a new free list. See questions 17 through 22. After initializing the pages (or immediately, if you respond no to initializing pages), the program requests a name for the new partition. See question 10.

The number of blocks per cylinder governs the size of a partition on a disk. All partitions must be multiples of the number of blocks per cylinder for a given disk. The minimum size for partitions is the number of blocks per cylinder, as follows:

- 35-megabyte Micropolis disk -- 120

- 24-megabyte Shugart disk -- 240

- 5.25" disks -- depends on the particular type and capacity of the disk

If you specify a number that is not a multiple of the number of blocks per cylinder, the next lowest multiple will be suggested.

10. Name of second half (new partition)? Partition name (up to 8 chars):

Enter the name of the new partition. The Partition program then asks if you want to initialize the pages for the new partition. See questions 18 through 22.

11. Do you want to merge this partition with the next? [No]

If you responded No to question 8 and the partition you named in question 7 was not the last

partition, this question allows you to combine two partitions. If either contains files and you wish to keep them, then both partitions must have disk relative free list hints and you must NOT initialize pages (question 20). Otherwise you will destroy the files if you merge. If in doubt, do a Debug run--this will tell you what kind of free list hints each partition has.

If you respond Yes to this question, you are asked to confirm your answer, then the program joins the partition you specified in question 7 with the partition which is next on the disk. This is the only time the order of the partitions on the disk matters. Partition then asks if you want to initialize the new partition. See questions 17 through 22. If you want to erase the new, bigger partition, respond Yes to question 20 to initialize the pages. To save all the files from both partitions, respond no to question 20 (but see the first paragraph above). Then, after Partition exits, run the Scavenger program on the new partition. When running the Scavenger in this case, be sure to tell it to rebuild the directories so that the directories of the two partitions can be joined together. You must not answer No to question 20 if either partition has partition-relative free list hints.

If you respond No to this question, Partition continues with the modification queries. See question 12.

12. Do you want to initialize this partition? [No]

If you responded No to question 11, this question allows you to initialize the partition specified in question 7.

Question 20 will ask whether or not you wish to initialize the partition pages.

If you respond Yes to this question, the next questions are 17 through 22. There are few reasons to initialize the partition without initializing its pages. However, if you respond No to question 20, you must use the Scavenger program to recreate the directory immediately after running Partition (but see Question 19).

If you respond No to this question, Partition continues with the modification queries. See question 13.

13. Do you want to change the partition name? [No]

If you responded No to question 12, this question allows you to change the name of the partition specified in question 7.

Before renaming a partition, note that the POS linker incorporates the name of the disk (device) and the partition where a program was linked into the Runfile. Therefore, if you rename the partition, no POS program in that partition can be run. Do not change the name of the partition that is used in the current path since this invalidates the default path name. Also, all entries in the search list that refer to the renamed partition will no longer work. After a rename, the POS system may not be able to find the shell or any other programs.

If you respond Yes, Partition requests confirmation and then prompts for the new partition name. See question 14.

If you respond No, the program exits (and has done nothing to the disk).

The default response to this question is No (to exit the program, rather than rename the partition).

14. New partition name [current name]

If you responded Yes to question 13 and to the subsequent request for confirmation, this question allows you to enter a new partition name. Enter a partition name from 1 to 8 characters. The

Partition program writes the name you enter in the Partition Information Block (PIB words 114 through 117).

Note that there is no default response to this question.

15. How many pages would you like in it? (0 => all, -1 => finish)

If you replied Yes to question 6, this question is asked for each partition in turn until the disk is full or you have reached the maximum number of partitions. The file system restricts disk partitions to fewer than 65,040 logical blocks (pages), however to be useable by POS there should be no more than 32520 blocks and there should be no more than 10 partitions in total in the first 31MB of the disk.

The number of blocks per cylinder governs the size of a partition on a disk. All partitions must be multiples of the number of blocks per cylinder for a given disk. The minimum size for partitions is the number of blocks per cylinder, as follows:

- 35-megabyte Micropolis disk -- 120

- 24-megabyte Shugart disk -- 240

- 5.25" disks -- depends on the particular type and capacity of the disk

If you reply -1 the sequence which started at question 6 is ended and there will still be some uninitialized blocks at the end of your disk. You may initialize these on a subsequent run of Partition.

If you specify a number that is not a multiple of the number of blocks per cylinder, the next lowest multiple will be suggested.

After you enter the number of blocks for a partition, the program requests the partition name. See question 16.

16. Partition name (up to 8 chars):

You must name each of the partitions (from one to eight characters). Examples of partition names used for the hard disk include "boot", "user", and "exp". A partition can have the same name as a disk, but all the partitions must have unique partition names.

See question 17.

17. Do you want to dismount the Partition? [Yes]

This question begins the partition initialization sequence. (You will not be asked this question if you are in -Debug mode.)

The usual reply is the default [Yes]. If you have any problems dismounting, e.g., if Accent could not mount the partition, you may reply No and then continue as normal.

If the last page of the partition is greater than 64K pages, the next question is 18; otherwise the next question is 19.

18. * Partition extends beyond 64K and so must use partition relative free list hints -- partition unusable by POS. OK to continue? [Yes]

The next question is 20.

19. Do you want partition relative free list hints? [No]

IMPORTANT - If you are not going to initialize the partition pages (answer NO to question 20)

for an existing partition, then you must be sure that you do not change free list hints from partition relative to disk relative or vice versa. Look at the partition information block summary.

Also you must always initialize pages when merging or splitting partitions if either or both have partition relative free list hints.

If you reply Yes, you will be asked to confirm your decision.

20. Next you will be asked if you want to initialize partition pages, and if so, whether you want to change the default interlace.

    a. Do you want to initialize partition pages? [Yes]

        Your response specifies whether or not the program initializes the partition's logical blocks (pages). If you are partitioning the remainder of the disk, that is, you responded Yes to Question 6, Partition repeats this question for each new partition on the disk. If you are modifying a partition (you responded No to question 6) and you want to initialize it (you responded Yes to questions 8, 11, or 12), Partition asks this question once for merge or initialize or twice for split.

        A Yes response to this question places all the logical blocks in the partition on the free list. If any logical blocks are found to be bad, they are removed from the free list and thus are never accessed again. Note that blocks that cannot be written are shown as an x on the display, but blocks in the bad block map (on the 5.25" disk only) are not written and are shown as B. Both are removed from the free list. If you are partitioning the remainder of the disk, that is, you replied Yes to Question 6), you should respond Yes to this question. If you are modifying a partition and initializing it (you responded YES to questions 8, 11, or 12), it is recommended that you respond yes to this question also. There are few reasons to initialize the partition itself without initializing the pages, and in general it is a dangerous thing to do (see question 19).

        If you respond Yes to this question, Partition asks whether you want to change the default interlace (Question 20b) and then whether or not to test after initializing the pages (Question 21).

        If you respond No to this question, you are asked to confirm your answer, and you are warned that you must use the Scavenger program to recreate the directory immediately after running Partition (but see Question 19).

        The default response is Yes (to initialize the partition pages).

    b. Do you want to change the default interlace? [16] [No]

        The usual reply is No. You may use this to optimize the speed of initializing your particular disk type.

21. Do you want to test after initializing? [Yes]

    You are asked this question if you replied Yes to question 20.

    Although testing after initializing slows the process somewhat, it is good practice to do this testing. If any logical blocks are found to be bad during testing, an x for each block is shown on the display and they are removed from the free list and thus are never accessed again.

    If you respond Yes, Partition asks if you want to write every page twice. See question 22.

    If you respond No, Partition simply initializes the logical blocks.

The default response is Yes (to test after initializing).

22. Do you want to write each block twice? [Yes]

You are asked this question if you replied Yes to question 21.

Writing every logical block twice provides some additional protection from bad pages; random data is written into the header and body of each block and then read back and compared. In practice, some bad blocks on the disk pass the first test and fail this one.

23. Do you want the partition remounted? [Yes]

In the current release, you will be asked this question only if you have not written to the disk (e.g., if you aborted before any writes were done). You will not be asked this question if the disk was written to (see Exit below) or if you're in Debug mode.

Only mounted partitions can be accessed, so if you plan to use the disk respond Yes.

24. EXIT

In the current release if you have written to the disk you will be asked to reboot.

5. Scavenger

The Scavenger provides disk maintenance. It checks all files for consistency, rebuilds the free list, and creates a new directory structure for the partition. The Scavenger also removes bad boots.

The Scavenger should be run only on devices initialized by the Partition program.

The maintenance provided by Scavenger may be required in the following situations:

- If you turned off the workstation without using the Bye command, the maintenance functions associated with the command were not performed. In a subsequent session the system may instruct you to run Scavenger.

- If a file was not closed properly because a program aborted, the directory in which it resides may be affected. If the file was a binary file, you will experience problems running the file. If the file was text, you will receive a message when you try to edit or type the file.

- When a partition is nearly full, the system may not have enough space to perform operations. If the system cannot perform for this reason it will ask you to delete some files and then run Scavenger.

You can run the Scavenger any time you suspect a file system problem.

Format:
```
SCAVENGER [/disk/partition]
```

If you omit the device and partition name, Scavenger asks a number of questions before it begins processing the partition. The questions and responses are discussed below. Note that some of the questions have a

default response. If you include a device and partition name, Scavenger uses all defaults and runs until completion unless there are any serious errors. If errors are discovered, Scavenger requests confirmation before exiting. Note that the partition you specify must be on the hard disk.

Do not type CTRL-DEL-c or CTRL-DEL-k after Scavenger begins writing on the device. (Either can be typed during read pass.) If you type either command after the Scavenger begins rebuilding the directory, you may not be able to access anything in the directory. If this happens, rerun Scavenger from another partition.

The Scavenger cannot recreate the directory if there are no free blocks in the partition. Therefore, if your partition is full, you must delete some files before running the Scavenger. If you cannot delete any files due to a bad directory and there are no free blocks, then you cannot rebuild the directory. In this case, you must initialize the partition (see Partition), thus losing all files there. Fortunately, this is a rare occurrence.

The Scavenger program fixes one partition at a time. You can scavenge the current partition or scavenge partitions other than the one you are currently running in.

If you are going to scavenge a 5.25" disk that you did not buy from PERQ Systems, you should add the following information to the Disk.Params file in your boot partition: disk name, number of heads, number of cylinders, write precompensation cylinder, boot size, and sectors per track. Otherwise, you will be required to provide this information each time you run Scavenger.

Phases

The Scavenger program has three separate phases. In phase one, it checks and updates some of the system information and deletes bad boots. If you define a boot with BindBoot, but either the microcode or system code files have been deleted, the boot is known to be bad and the Scavenger deletes it.

In the second phase, the Scavenger checks the partition you specify for consistency. The program reads all blocks and generates a new free list in ascending disk order (the Scavenger discards the old free list). Also during this phase, the Scavenger marks as "bad" those blocks that are not readable. If the blocks marked as bad cannot be rewritten, they are marked as "incorrigible" and removed from the file system. All blocks that were allocated to files containing bad or incorrigible blocks are put in a bad file. Additionally, malformed chains are added to the bad file. You specify a name for this bad file in phase three.

During phase three, the Scavenger rebuilds the directories for the partition. You can direct the Scavenger to delete the old directories, if desired. Otherwise, the old directories are marked as such and the program adds a dollar sign to the end of the directory name. The Scavenger aborts if there is not enough room in the partition to create copies of all the directories, leaving the directories only partially rebuilt. In this case, delete some of the files in the directory and then rerun the Scavenger.

27 August 1984

Prior to entering a name in a directory or creating a new directory, the Scavenger validates the filename. If a bad name is found (the name includes a wildcard, "/", ":", or is null) or two files have the same name, Scavenger requests a new filename. In addition, the name cannot end with a "/" or contain "/../" or "/./". After Scavenger is finished, you can examine the files with bad names to see whether they contain useful information. If so, rename or edit the files to recover the data. Otherwise, simply delete the files.

Also during the third phase the Scavenger checks the length of all files and allows you to specify a new length. Note that the length refers to the stored length (in the FIB) rather than the number of blocks in a file. Certain files, like directories and swap files, do not set the length field. File lengths usually become garbled when the file is opened and written, but not properly closed (for example, when a transfer is aborted).

Questions

Following is a list of the questions Scavenger will ask, with an explanation of each.

1. Which unit to scavenge [Sys]?

   In response to this question, specify the device that contains the partition you wish to modify.

   Scavenger checks to see what type of disk drive you are using and asks whether your disk is a MICROPOLIS 8-inch drive, a Shugart 14-inch drive, or a 5.25-inch drive. See question 2, 3, or 4.

2. This seems to be a MICROPOLIS 8-inch disk. Is this right? [Yes]

   This question confirms the disk choice.

   The next question is number 6.

3. This seems to be a SHUGART 14-inch disk. Is this right? [Yes]

   This question confirms the disk choice. Scavenger then checks to see whether the disk is a 12- or 24-megabyte disk and asks for confirmation of the size chosen. See question 5.

4. This seems to be a 5.25" disk. Is this right? [Yes]

   This question confirms that a 5.25-inch disk is being used. The next question is 6.

5. Is this a nn-MByte disk? [Yes]

   This question confirms the size of the Shugart disk. The default is the size of disk. In the actual question, the size (12 or 24) replaces nn above. All CMU Shugart disks are 24 MBytes.

   The next question is number 6.

6. Can I make changes to your disk [Yes]?

   The response to this question specifies whether or not the Scavenger program can make changes to the device in the first two phases of the program (the Scavenger fixes directories later). This is similar to the debug option in the Partition program (refer to question 1 under Partition).

   If you respond No, the Scavenger checks the partition for errors and reports them, but does not fix anything. If you are running the Scavenger only to fix the directory, it is about twice as fast to respond No to this question; otherwise, Yes is a good idea.

   The default is Yes. You are asked to confirm a Yes response.

7. Do you want logical block consistency checking [Yes]?

The logical header of every block contains information about the state of that block. The information includes the count of the blocks in the file (the file relative LBN) and a two-word identifier for the file the block belongs to (the serial number). This question and the next one allow you to check the correctness of these values.

This question asks whether or not the Scavenger should check the Logical Block Number field in the block's logical header. Remember that this field specifies the file relative logical block number (the first, second, third, and so on block of the file).

If you respond Yes, the Scavenger compares the LBN field in the logical header with the random index for each file.

If you respond No, the Scavenger omits this test.

The default is Yes (to check the file relative logical block number).

8. Do you want serial number consistency checking [Yes]?

This question asks whether or not the Scavenger should check the Serial Number field in the block's logical header. The serial number of all blocks in a given file is the Logical Disk Address of the file's File Information Block.

If you respond Yes, the Scavenger compares the Serial Number field in the logical header with the LDA of the FIB.

If you respond No, the Scavenger omits this test.

The default is Yes on workstations with more than 256k bytes of main memory and No on 256k byte workstations.

If the Scavenger continually aborts due to FullMemory, respond No to this question. The default response of No for serial number checking on workstations with 256k bytes of main memory avoids the FullMemory condition; simply press return to get the default answer.

9. How many tries for a suspect read? [1]

This question asks for the number of retries for a suspect read. The default is 1.

If you responded Yes to question 6 (whether or not the Scavenger could change your disk), the Scavenger asks questions 10 through 13.

If you responded No to question 6, the Scavenger asks question 14.

10. Do you want temporary segments deleted [Yes]?

This question asks whether or not the Scavenger should delete temporary files. Temporary files exist for swapping; all user files are permanent.

The recommended response is Yes, which is the default.

11. Can I rewrite incorrigible blocks [No]?

This question asks whether or not Scavenger can rewrite incorrigible blocks. If a block cannot be successfully read or written in the specified number of retries (see question 9), writing new data onto the block could fix the problem. However, because writing new data has only a small likelihood of fixing the problem, the default response is No.

If you respond Yes, the Scavenger writes new data in the incorrigible blocks. If the write or a subsequent read fails, then the block is "incorrigible," otherwise it is "bad."

If you respond No, the block is marked "incorrigible" as soon as the read fails. If there are only transient read errors, the block is left alone.

12. Type pairs to ignore (cyl head cyl head ...): []

This question permits you to enter cylinder/head pairs that you assume are bad.

If you suspect a bad cylinder or track, enter the cylinder number followed by a space followed by the head number. Continue this sequence for the suspect cylinder/head pairs. Note that a carriage return signifies end of input. Therefore, if the bad pairs require more than a single line for input, simply wrap the pairs to the next line; do not press carriage return until you enter all the bad pairs.

The default ignores no cylinder/head pairs.

13. Type other blocks to ignore: []

This question permits you to enter block numbers that you assume are bad.

If you suspect a bad block, enter the Logical Block Number of the bad block. You can enter up to 15 bad blocks, separated by spaces. Again, carriage return signifies end of input. If you suspect that more than half the logical blocks are bad on a track, writeoff the entire track (via question 12). Disks have the following number of blocks per track: Shugart (either a 12M or 24M), 30; Micropolis, 24; and 5.25" disk, 16.

The default ignores no logical blocks.

Scavenger then asks whether or not you want complete error listing; see question 14.

14. Do you want complete error listing [Yes]?

This question permits you to specify whether or not you want notification of individual errors.

Respond Yes for complete error listing or No for a summary of errors. Yes is the default.

After you answer the above questions, the Scavenger begins fixing the partition.

The Scavenger updates the title line of the window to show what it is working on.

First, the Scavenger fixes discrepancies if you allowed changes (responded Yes to question 6). If the Scavenger finds a problem with the partition or device information blocks it cannot fix, it asks for help. The problem could be that you specified the wrong device type in question 1 or that the device is not a file system device (for example, an RT-11 format floppy or a hard disk that has not been initialized by the Partition program). These are easily remedied. However, the problem could be that the device discrepancies are beyond repair. Unfortunately, the only fix in this case is to re-partition the device from scratch.

After the device and partition information checks out, Scavenger displays a summary of the Device Information Block (including partition names) and asks which partition it should work on.

15. Which partition do you want to scavenge?

Enter the name of the partition to scavenge. There is no default response to this question.

When you enter the partition name, the Scavenger displays a summary of the specified partition's Partition Information Block. Note that the values are those stored in the PIB prior to the scavenge.

The Scavenger now makes a read pass through the partition, building tables of each block's next and previous link. The pass is done in eight parts for efficiency.

The Scavenger then checks the tables built during the read pass for consistency. If any loops are found, the Scavenger breaks the loops and blinks the screen to show that a loop has been fixed.

Following the consistency check, the Scavenger rebuilds the free list. Note that the Scavenger rebuilds the free list only if you responded Yes to 6 (you permitted changes to the disk). Rebuilding the free list requires a write pass for all blocks on the free list. If any bad blocks were found, some more reads and writes are necessary to make the bad blocks into a well-formed chain.

At this point, the Scavenger writes the new Partition Information Block and then begins the directory building pass. The Scavenger asks whether it should rebuild the directory.

16. Do you want to rebuild the directories?

This question permits you to rebuild the partition's directory structure.

Sometimes the Scavenger recommends that you do this, in which case the default is Yes. Otherwise the default is No.

If the Scavenger recommends that you rebuild the directories or if you suspect there is a problem with the directories and there are enough free blocks, respond Yes. If you respond "No" because there are not enough free blocks, you should delete some files and run scavenger again. This time you should rebuild directories whether Scavenger suggests it or not.

If you respond Yes, the Scavenger asks for confirmation, and then asks if you want to delete the old directories (see question 17).

17. Delete old directories [Yes]?

If you responded Yes to question 16, this question permits you to delete the old directories.

If you respond Yes, the Scavenger deletes the old directories.

If you respond No, the Scavenger saves the old directories for later inspection; the Scavenger appends a dollar sign to the end of the old directory names and changes their file type to ExDirFile (directories all have the type DirFile). The old directories are just files that you can delete after the scavenge.

The Scavenger then creates new directories as needed. Note that empty directories are not recreated.

Since the directory reappears after a scavenge, the directory rebuilding process makes it easy to recover from overwriting or deleting a directory.

The Scavenger checks and allows fixing file lengths if desired. For each file, it checks the stored length with the actual number of blocks in the file. If they do not match, it allows you to specify a new stored length. This can be any value, but making it bigger than the number of blocks in the file is not recommended. The default for the stored length is the number of blocks in the file. The Scavenger does not check the lengths for directory files or files with their type field set to "SWAPFILE."

Each file's random index allows the file system to find a random logical block without searching down the chain from the file start. The Scavenger, as part of the directory building phase, can rebuild the random indices for all files. There is a separate question for this with a default answer of No. There is usually no reason to rebuild the indices unless Scavenger asks you to. Building the random indices for large files takes a long time.

If a bad file was created, the Scavenger asks for a name for the file at the end of the directory building phase. If you allow Scavenger to enter and fix the indices for the bad file, you can then

type and edit it as a normal file. In this way, you can reclaim some useful information, by copying. Do not delete or rename this file since that will likely cause need to run Scavenger again and possibly lose other files.

18. Do you want to remount the partition [Y]?

Only mounted partitions can be accessed, so if you plan to use the disk respond Yes.

Index